Restricted Set Classification with prior probabilities: A case study on chessboard recognition

Kuncheva, Ludmila; Constance, James

Pattern Recognition Letters

DOI:
10.1016/j.patrec.2018.04.018

Published: 01/08/2018

Peer reviewed version

Restricted Set Classification with prior probabilities: A case study on chessboard recognition

Ludmila I. Kuncheva*, James H. V. Constance

School of Computer Science, Bangor University, Dean Street, Bangor Gwynedd, LL57 1UT, United Kingdom

Abstract

In the Restricted Set Classification approach (RSC), a set of instances must be labelled simultaneously into a given number of classes, while observing an upper limit on the number of instances from each class. In this study we expand RSC by incorporating prior probabilities for the classes and demonstrate the improvement on the classification accuracy by doing so. As a case-study, we chose the challenging task of recognising the pieces on a chessboard from top-view images, without any previous knowledge of the game. This task fits elegantly into the RSC approach as the number of pieces on the board is limited, and each class (type of piece) may have only a fixed number of instances. We prepared an image dataset by sampling from existing competition games, arranging the pieces on the chessboard, and taking top-view snapshots. Using the grey-level intensities of each square as features, we applied single and ensemble classifiers within the RSC approach. Our results demonstrate that including prior probabilities calculated from existing chess games improves the RSC classification accuracy, which, in its own accord, is better than the accuracy of the classifier applied independently.

1. Introduction

Restricted set classification (RSC) refers to the following problem. Given is a set containing m instances, $X = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$, where $\mathbf{x}_j \in \mathbb{R}^n$, $j = 1, \ldots, m$, is a data point in some n-dimensional space. Each instance must be labelled in one of c classes from the set $\Omega = \{\omega_1, \ldots, \omega_c\}$. It is known that the maximum number of instances from class ω_i, present within X, is k_i, $i = 1, \ldots, c$. Thus the cardinality of X must satisfy $1 \leq |X| \leq \sum_{i=1}^{c} k_i$.

The solution to this problem is not straightforward. If a classifier is trained and then applied for labelling the instances in X (called the 'independent classifier'), the obtained labels are not guaranteed to meet the count constraints. Incorporating these constraints into the classification process has been shown to lead to an improvement on the accuracy of the independent classifier [14, 15]. Here we hypothesise that a further improvement can be achieved if prior probabilities depending on the whole of X are considered by the RSC set classifier.

Examples of real-life RSC problems include recognising people in a group (e.g., for attendance monitoring of the students in a class [1] or for tracking [17]) and identification of animals for the purposes of monitoring and conservation [13, 7]. A particularly suitable application is identifying the pieces on a chessboard from an image. When classifying chess pieces *together*, we can take advantage of the knowledge that there can only be a given number of objects from each class. For example, there can be at most eight white pawns on the board. In this paper, we chose chessboard recognition as an example to demonstrate the expected improvement on the classification accuracy of the independent classifier when using prior probabilities.

The rest of the paper is organised as follows. The RSC approach is detailed in Section 2. Our proposed extension is described in Section 3. Section 4 contains our case-study which demonstrates the improvement of the proposed approach over the original RSC in recognising chess pieces on a board. Section 5 offers our conclusions and ideas for future work.

2. Restricted Set Classification (RSC)

RSC is detailed in Algorithm 1. The RSC approach operates by applying a pre-trained classifier D to X to acquire estimates of the posterior probabilities for every instance within, and making an optimal label assignment while observing the count restriction. The classifier D is termed *the independent classifier* as it is trained on independent, identically distributed (i.i.d.) data, and is oblivious to any count limits. This can be any classifier which returns estimates of the posterior probabilities, $D(\mathbf{x}) = \{P_D(\omega_1|\mathbf{x}), \ldots, P_D(\omega_c|\mathbf{x})\}$. Denoting the space of probability distributions over Ω by $\mathcal{P}(\Omega)$, we have $D : \mathbb{R}^n \rightarrow \mathcal{P}(\Omega)$. It is desirable that these estimates are well calibrated [5].

D can be a single classifier or a classifier ensemble itself, as long as the output is a probability distribution. Straightforward estimates of the posterior probabilities from a classifier ensemble are the proportions of votes for the respective classes.

The posterior probability estimates for all instances in X are organised in an $m \times c$ "probability matrix" P^p, where row i represents the probability distribution obtained from D for instance $\mathbf{x}_i \in X$. Subsequently, an augmented probability matrix, P^a is

*Corresponding author.
Email address: l.i.kuncheva@bangor.ac.uk (Ludmila I. Kuncheva)

constructed by repeating each column of P^p as many times as the number of allowed instances from the corresponding class. For example, if $k_1 = 3$ and $k_2 = 4$, the first three columns of P^a will be copies of the first column of P^p, followed by four copies of the second column of P^p. Thus the size of P^a is $m \times q$, where $q = \sum_{i=1}^{c} k_i$. We have previously proved [15, 14] that the optimal assignment guaranteeing the minimum Bayes error in labelling the whole of X requires that the product of the posterior probabilities is maximum, that is

$$\langle \omega_1^*, \omega_2^*, \ldots, \omega_m^* \rangle = \arg \max_{\langle \omega_1^{(a)}, \omega_2^{(a)}, \ldots, \omega_m^{(a)} \rangle} \prod_{i=1}^{m} P\left(\omega_i^{(a)} | \mathbf{x}_i\right) , \quad (1)$$

where $\omega_i^{(a)}$ is the class label assigned to $\mathbf{x}_i$, and ω_i^* is the optimal label. This optimisation must be carried out subject to the condition that the number of labels for class ω_j in the returned set must be no greater than the restriction constant k_j, $j = 1, \ldots, c$. The construction of the augmented matrix with posterior probabilities guarantees the compliance with the constraints. In order to find the optimal $\langle \omega_1^*, \omega_2^*, \ldots, \omega_m^* \rangle$, we need a matching procedure. The Hungarian algorithm finds the optimal match which *minimises* the *sum* (or cost) of assignments. Therefore, in order to use this algorithm we convert the product in eqn. 1 into a sum of logarithms. As we are seeking to *maximise* this sum while the aglorithm looks for minimum cost, we submit to the Hungarian algorithm the matrix with the negative logarithms P^a.

Algorithm 1: Restricted Set Classification

Input: Pre-trained classifier $D : \mathbb{R}^n \to \mathcal{P}(\Omega)$, the allowed number of instances from each class $K = \{k_1, \ldots, k_c\}$, a set of instances to be classified together $X = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$, $\mathbf{x}_i \in \mathbb{R}^n$.

Output: Labels L for the instances in X.

```
// acquire probability matrix P^p
for i ← 1,...,m do
    P^p(i, 1:c) ← D(x_i)
// construct augmented probability matrix P^a
P^a ← ∅.
for i ← 1,...,m do
    cc ← 1 // column counter
    for j ← 1,...,c do
        for k ← 1,...,k_i do
            P^a(i, cc) ← P^p(i, j)
            cc ← cc + 1
// find optimal label assignment M
M ← hungarian-assignment(−log(P^a))
L ← retrieve-labels(M)
Return L.
```

The output of the Hungarian algorithm is a binary matrix M of the same size as P^a ($m \times q$), containing 1s where rows are assigned the column label, and 0s elsewhere. Each row (instance in X) has one and only one assigned column. The class label of the instance is retrieved by identifying which class label has given rise to the column in P^a. In the above example, if a column between 1 and 3 contains the 1 for the row, the label for the instance is ω_1. Alternatively, if the 1 is in one of the columns between 4 and 7, class ω_2 will be retrieved.

The theoretical grounds and empirical evidence that the RSC works better than m independent applications of D to the elements of X are given in the original work [15]. Here we are interested in extending RSC to incorporate prior probabilistic information, as proposed next.

3. Incorporating a conditional prior into RSC

Suppose that by analysing a large prior database, we were able to obtain prior probabilities depending on some parameter of the set of instances X. This parameter can be, for example, the cardinality of X or some relationship between the instances in X, $\theta = \theta(X)$. Say, we are recognising the students in a class from a photo of the classroom. While the students can sit wherever they choose in the classroom, some usually pick the same seats. We can use a parameter such as

θ = Sitting in the first row? (y/n),

and pre-calculate a prior probability for each student (class) conditioned on θ. The appearance of the student's face in the photo, which would be their feature vector $\mathbf{x}$, will not depend on θ.

Denote by $P_P(\omega_i|\theta)$ the conditional prior probability for class ω_i, $i = 1, \ldots, c$. To integrate this probability within the probabilities obtained from the independent classifier, P_D, we use

$$P(\omega_k|\mathbf{x}, \theta) = \frac{P(\mathbf{x}, \theta|\omega_k) P(\omega_k)}{P(\mathbf{x}, \theta)}$$

Assuming independence between $\mathbf{x}$ and θ,

$$\begin{aligned} P(\omega_k|\mathbf{x}, \theta) &= \frac{P(\mathbf{x}|\omega_k)\ P(\theta|\omega_k)\ P(\omega_k)}{P(\mathbf{x})\ P(\theta)} \\ &= \underbrace{\frac{P(\mathbf{x}|\omega_k)\ P(\omega_k)}{P(\mathbf{x})}}_{\text{posterior}} \frac{P(\theta|\omega_k)}{P(\theta)} \end{aligned}$$

Multiplying and dividing by $P(\omega_k)$,

$$\begin{aligned} P(\omega_k|\mathbf{x}, \theta) &= P(\omega_k|\mathbf{x})\ \frac{P(\theta|\omega_k)\ P(\omega_k)}{P(\theta)}\ \frac{1}{P(\omega_k)} \\ &= P(\omega_k|\mathbf{x})\ \frac{P(\omega_k|\theta)}{P(\omega_k)} . \end{aligned}$$

Any estimate of the probabilities can be plugged in this equation. In our case:

$$P_E(\omega_k|\mathbf{x}, \theta) = \underbrace{P_D(\omega_k|\mathbf{x})}_{\text{from } D} \quad \underbrace{\frac{P_P(\omega_k|\theta)}{P_P(\omega_k)}}_{\text{from the prior database}} .$$

We may wish to control the influence of the conditional prior probability on the final posterior probability. Therefore we introduce a tunable parameter, $\beta \in [0, 1]$, as follows:

$$P_E(\omega_k|\mathbf{x}, \theta) = P_D(\omega_k|\mathbf{x}) \left[\frac{P_P(\omega_k|\theta)}{P_P(\omega_k)} \right]^{\beta} . \quad (2)$$

This probability distribution across the class labels Ω should be calculated for each instance $\mathbf{x}_j \in X$ and used instead of $D(\mathbf{x}_j)$ in constructing P^p in Algorithm 1.

Note that the conditional prior is only available in relation to the whole set X. Arguably, this probability extension can be thought of as coming from an extra classifier built upon an alternative feature space containing only θ.

At this point, several questions may arise: What kind of class priors should be used? How could the choice of such priors impact the overall performance of the proposed extension? In theory, adding a new feature (in this case θ) to a given classifier model cannot harm the performance, but can improve it. Our model of including this new feature depends on two factors: the independence assumption holding, and the accuracy of the approximation of the probabilities of interest $P_P(\omega_k|\theta)$ and $P_P(\omega_k)$.

For the task of chess piece recognition, our θ is the number of pieces on the board and the position of the square. We can assume that θ and the appearance of the image of a square $\mathbf{x}$ do not depend on one another as the photographs are taken in the same way regardless of the number of pieces. Then the deciding factor is the accuracy of approximation of the two probabilities. While $P_P(\omega_k)$ is easy to obtain even from a small number of boards, $P_P(\omega_k|\theta)$ requires a lot more data. In order to find a reasonably accurate approximation for each square, given the total number of pieces on the board, a large number of boards must be available. Note that the "heavy-duty" data collection in our case is acquiring the images of the boards and the squares, while estimating the probabilities from hundreds of thousands of recorded games requires only a simple calculation. The availability of large databases of historical chess game records gave us the idea to exemplify the RSC extension by chess piece recognition. We investigate in Section 4.6 the sensitivity of our method to the number of data samples from which $P_P(\omega_k|\theta)$ is calculated.

In this paper we raise a new hypothesis: by including prior probabilities in the RSC, the obtained set classifier will be significantly better than that without the prior probabilities, and also significantly better than the independent classifier, with or without the conditional prior probability. While a single example cannot corroborate the overall validity of our hypothesis, below we provide support for it through a case study.

4. A case study: Recognising chess pieces on a board

While online chess games are played using a representation of the board on the screen, rated games are usually played in the traditional way, face to face, over a physical, three-dimensional board. Most commonly, games are recorded by the players on a piece of paper as they play, and at the end of the game the moves are manually entered into chess software. For some professional games, such as the World Championships or high-level tournaments, the games are played on DGT (Digital Games Technology) electronic boards [23], which can sense the identity and location of pieces on the board. However electronic boards are expensive. A cheaper alternative would be to use images or video-feed from a standard physical chessboard.

4.1. Related Work

Since the late 18th century, when a fake chess-playing machine called The Turk, (the Mechanical Turk or the Automaton Chess Player) was introduced to the Empress Maria Theresa, robotic chess-players have attracted the attention of the public and researchers alike. A full-scale design of playing robot relying on machine vision must address the problems of the physical piece movement in addition to the processing of the video-feed, recognising the move of the opponent, and querying a chess engine to identify the best move that the robot should make [6, 16, 22, 4].

Almost invariably, the systems for chess board and piece recognition are based on image difference. A pair of images is acquired, one "before" the move, called the *reference* image, and one "after" the move. The difference is used to identify a region-of-interest in the image. Combined with the knowledge of the piece positions before the move, and the possible legal moves, this approach makes the identification problem much easier than using a single image and no history of the game progression.

Depending on the physical set-up, different approaches have been proposed for piece recognition. Cour et al. [6] use an overhead webcam to track the moves, as do Wang and Green [24], Koray and Sümer [12], and Illeperuma [10], while Chen et al. [4] view the board from a camera held at a small angle from the vertical. Piškorec et al. [21], on the other hand, use an overhead camera to track the moves in conjunction with a second camera with a side-view to identify piece types. The current consensus is that a top-view camera is of no use for piece recognition [19, 21, 9]. Side views contain information about the piece silhouettes, which is deemed more suitable for the task [9, 21]. The features (descriptors) are typically the Fourier coefficients of the cumulative angular function of the shape [25]. A problem with this approach is that pieces may not be clearly visible, hence occlusion by other pieces may need to be taken into consideration. Schwenk and Yuan [22] create 3-D models of the shapes, and subsequently render and project a 3-D image of a piece onto the board. They match the projection to the side board view, and choose the pieces whose projection matches the current image most closely. Often, the board is modified in order to allow for identification of the squares which does not interfere with the piece recognition. For example, red and green colours are chosen for dark and light squares, instead of the standard equipment used in chess tournaments [9, 21, 10].

The shortcomings to the *tracking* approach are as follows: (1) the initial position must be known, and manually entered in the tracking system; (2) if a move is misidentified, subsequent positions will be affected, propagating the error.

As opposed to tracking the position of the pieces, there is very little literature on piece recognition from a single static image. The silhouette approaches could be useful but the occlusion problem may prove a significant obstacle without knowledge of the previous position of the board. If the image is taken at an angle which favours the silhouettes, the information about the square occupancy may become insufficient or unreliable. In this paper, we take the task of identifying the whole chess board from a top-view image, without any knowledge of the game moves leading to the current position.

4.2. Chess piece recognition as an RSC problem

There are 13 classes of chess pieces as shown in Table 1. The independent classifier D will be trained with cropped images of squares from the chessboard. Each square is pre-labelled into one of the 13 classes. Table 1 also shows the maximum allowed number of pieces (k_i) for each class.

The set X contains 64 images of squares which make up a whole board.

The parameter θ in our case study consists of two components: the (estimated) number of pieces on the board, and the position of the square in the board. To incorporate this information, we treat both the board position and the number of pieces as nominal variables, and use a look-up table with precalculated probabilities.

Denote by t the number of pieces on the chess board, $t \in \{2, 3, \dots 32\}$, and by r the position of a square, $r \in \{1, 2, \dots 64\}$. Then $P_P(\omega_i|\theta) = P_P(\omega_i|t, r)$ is the conditional prior in eqn (2).

Since we aim to recognise one whole board at a time, X contains all the squares in the board ($|X| = 64$), and the size of the augmented matrix in Algorithm 1, P^a, is 64×94.

The number of pieces is not immediately available for a given X. We assume that distinguishing empty from non-empty squares is an easier task compared to any piece recognition. Therefore we take the estimate of the number of pieces t from our classifier applied on X.

It is difficult to advise a researcher on the choice of θ for their specific problem. In fact, the availability of large number of past records of chess games guided us to choose the number of pieces for each square as a prior. This choice came before the statistical workout. Suitable candidates for the priors' parameter would be easily obtainable features from large existing databases or the Internet, for example using web-priors for video summarisation [11].

4.3. Organisation of the experiment

4.3.1. Data

We used a collection $\mathcal{D}$ of 3583 games from edition 1144 of The Week In Chess Magazine, published online on 10/10/2016[1].

The data was divided into three parts.

• We sampled 100 boards from $\mathcal{D}$ as the training data, $\mathcal{D}_2$, which we use to explore the influence of the tunable parameter β in eqn. (2) on the accuracy of the extended RSC.

• Another set of 100 boards was sampled from $\mathcal{D}$ as the testing data, called $\mathcal{D}_3$.[2] This data set was not seen at any stage of the training. Using the parameter values chosen on $\mathcal{D}_2$, and training a classifier on the whole of $\mathcal{D}_2$, we subsequently tested all accuracies of interest on $\mathcal{D}_3$.

• The reminder of $\mathcal{D}$, after removing $\mathcal{D}_2$ and $\mathcal{D}_3$, was taken forward as $\mathcal{D}_1$, from which we calculated the prior probabilities. Note that data $\mathcal{D}_1$ is not used for anything else during training and testing.

The 200 boards for $\mathcal{D}_2$ and $\mathcal{D}_3$ were arranged on a physical board and photographed from above. The top-view images of the chessboards were processed to separate the individual squares on each board. An example of the colour-enhanced image of a board with the square corners marked with green x is shown in Figure 1. The inner 7-by-7 grid-points were detected[3] and were subsequently augmented with the outer grid points to achieve the segmentation shown in the figure.

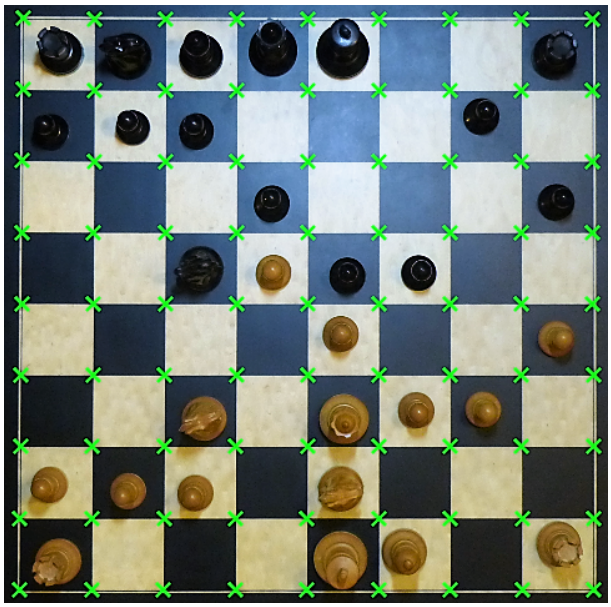

Figure 1: Top-view image of the chessboard with segmented squares.

Examples of the acquired images for sets $\mathcal{D}_2$ and $\mathcal{D}_3$ are shown in Figure 2. The only features we considered first were the grey level intensities of the pixels. Each square was resized to a given resolution and the pixel intensities were concatenated. For example, for a 10-by-10 resolution, each square was represented by 100 features (pixel intensities).

The majority of our experiments are carried out with this most basic set of features for at least two reasons: (1) to make the classification task reasonably difficult so that we can showcase the difference between the proposed and the standard solutions; (2) to ensure that our experiments would be easily reproducible by a non-expert. We further experiment with more advanced feature representations, as reported in Section 4.5.

[1] `http://theweekinchess.com/twic`

[2] The indices of the sampled boards are available in MATLAB format from `https://github.com/LucyKuncheva/Chess-piece-recognition`

[3] MATLAB function `detectCheckerboardPoints` was used.

Table 1: Classes in the chess-pieces recognition problem, and the limit number for each class in a standard chess game.

Class #:	1	2	3	4	5	6	7	8	9	10	11	12	13
	king	queen	rook	bishop	knight	pawn	king	queen	rook	bishop	knight	pawn	empty
Numbers: allowed	1	1	2	2	2	8	1	1	2	2	2	8	62

We experimented with squares with the following resolutions: 5×5, 10×10, 25×25, and 50×50.[4]

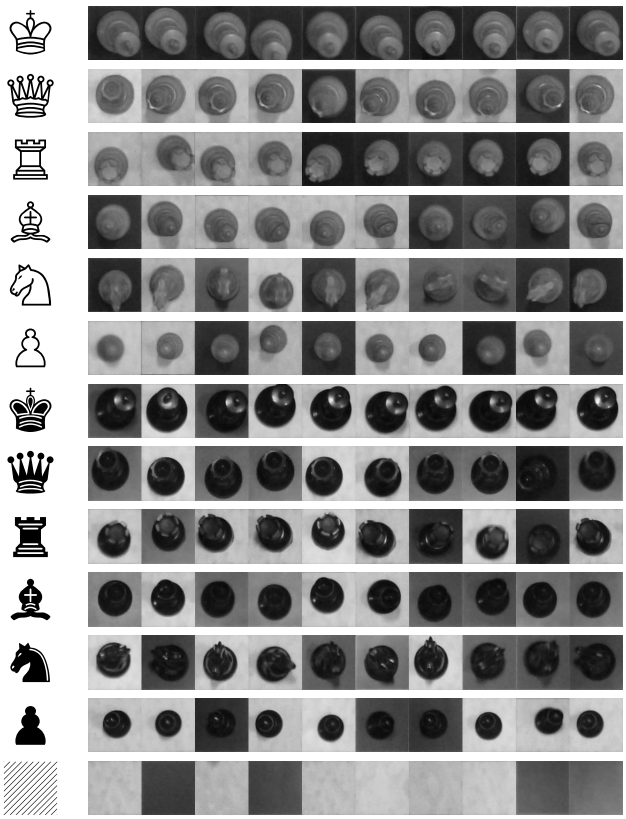

Figure 2: Examples of the images of the 13 classes of squares on the chessboard.

4.3.2. The independent classifier

Bearing in mind that the output of the independent classifier must be estimates of the posterior probabilities for the classes in Ω, we chose the following collection of models for D:

1. A customised nearest neighbour classifier (c-1nn) which returns posterior probabilities based on the distance between the instance $\mathbf{x} \in \mathbb{R}^n$ and its nearest neighbour from each class. Then the posterior probability for class ω_i is estimated through the softmax rule:

$$P_D(\mathbf{x}|\omega_i) = \frac{\exp(-d_i)}{\sum_{j=1}^{c} \exp(-d_j)},$$

where d_i is the distance between $\mathbf{x}$ and its nearest neighbour among the reference points from class ω_i.
2. Bagging classifier ensemble with decision trees as the base classifier [2]. We set the number of classifiers to 200. The posterior probabilities are calculated as the proportion of individual classifiers which vote for the respective class. This calculation is the same for all ensemble methods used as D.
3. Random Subspace classifier ensemble with 1-nn as the base classifier. Twenty features were sampled for each base classifier.
4. Random Forest ensemble with 200 classifiers. [3]

4.3.3. Experimental protocol

Using $\mathcal{D}_1$, we calculated the look-up table of size $31 \times 64 \times 13$ (number of pieces[5], number of squares, number of classes). Entry (i, j, k) in the table is an estimate of

$$P_P(\omega_k | t = i + 1, r = j),$$

where t is the number of pieces on the board, and r is the position of the square out of the 64 possible positions. To calculate this estimate we first located the position of the square, and then took the number of occurrences of each piece in this square. The proportion of occurrences of a given piece was taken as the estimate $\hat{P}_P(\omega_k | t = i + 1, r = j)$.

The prior probabilities $P_P(\omega_i)$ were calculated from $\mathcal{D}_2$ as the proportion of the classes.

Next we ran a training cycle to determine the best parameter value, choosing among: $\beta = \{0.01, 0.03, \ldots, 0.19\}$. For each classifier model, we ran a 100-fold cross-validation on $\mathcal{D}_2$. Each fold was a complete chessboard containing 64 instances. The reason for this choice is that we are interested in two measures of the accuracy with respect to X [15]:

- A_P, partial accuracy: A_P is the proportion correctly labelled instances in X (the conventional estimate of the classification accuracy).

[4] The images of the training and testing data with resolution 100×100 are provided in GitHub `https://github.com/LucyKuncheva/Chess-piece-recognition`.

[5] Minimum possible number of pieces left on the board is 2, and maximum is 32.

- A_T, total accuracy: $A_T = 1$ if all labels are correctly assigned to the instances in X, and $A_T = 0$, otherwise.

For every fold, we calculated the pair of accuracies (A_T, A_P) for the following scenarios, where the testing set X was the chessboard left outside the training:

1. D, the independent classifier applied on its own.
2. $D+P$, where P stands for 'conditional prior'. The label for $\mathbf{x}$ is assigned by the maximum $P_E(\omega_k|\mathbf{x}, t, r)$, $k = 1, \ldots, c$, as in eqn. (2).
3. RSC(D), where the restricted set classification model is applied only with the independent classifier D, as in [15].
4. RSC($D + P$), which is the proposed extension.

For each classifier model, we chose β which maximised A_T *for the respective scenario*. These parameter values were taken forward for the classification of the testing data $\mathcal{D}_3$.

Finally, we applied the 4 classifier models of D and the 4 scenarios to $\mathcal{D}_3$. Each of the 100 chessboards was considered as X, and the two accuracies (A_T, A_P) were calculated.

As we are interested in the difference between the proposed extension RSC($D + P$) and the non-extended versions, we ran a statistical test. The null hypothesis H_0 was that there is no difference between the mean accuracy (A_T or A_P) of RSC($D + P$) and the chosen rival among the other three scenarios. The alternative hypothesis H_1 was that the mean accuracy of RSC($D+P$) is higher than that of the rival scenario. As all accuracies are commensurable, we ran a right-tailed paired t-test.

For comparing the paired A_T scores for $\mathcal{D}_3$, we note that they are collections of 100 binary values, which we can interpret as true (all squares on the board labelled correctly) and false (there has been at least one mistake). The statistical test suitable for this type of data is the McNemar test [18]. The null hypothesis H_0 for this test is that there is no difference between the proportion of 1s in both candidate sets. The alternative hypothesis H_1 is that the proportions are not equal (the difference could be in either direction).

4.4. Results

Figures 3 and 4 show respectively accuracies A_P and A_T for the four models chosen for the independent classifier D as functions of the image resolution. All 4 scenarios are plotted in every diagram. Red lines and small triangle markers indicate that RSC is used, while black lines and square markers indicate that only an independent classifier is used, be it with or without modifications. Solid lines show the scenarios where priors are used, and dashed lines, the scenarios without the proposed extension. Large triangle markers indicate no significant difference from the RSC($D + P$) point at significance level 0.05 using paired right-tailed t-test. For all other points below the RSC($D + P$) graph, the difference is statistically significant.

It can be seen that, in both figures, the lines for RSC($D + P$) are above the lines for the rival scenarios, which demonstrates the advantage of our proposed extension of RSC. The patterns for all examined classifier models are very similar, indicating the robustness of the proposed extension. The proposed extension is also not sensitive to the image resolution; it dominates the rival scenarios for all four resolutions.

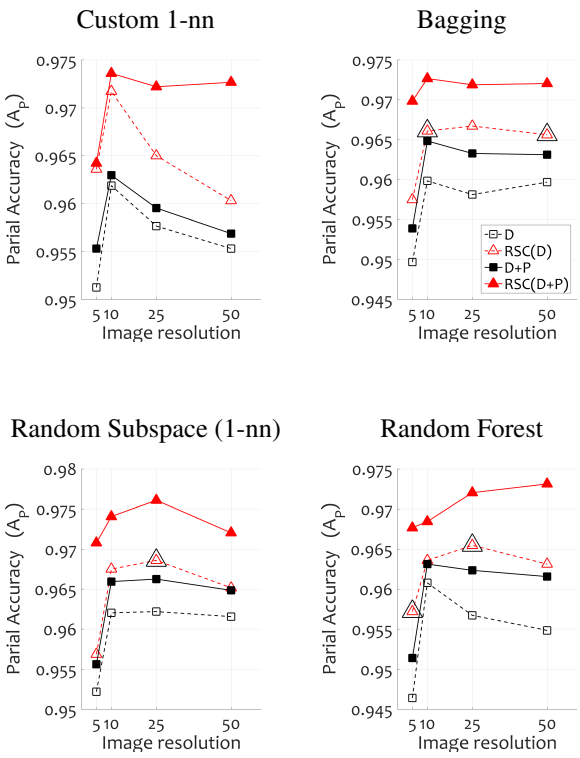

Figure 3: *Partial* accuracy (standard classification accuracy) A_P for the 4 independent classifier models. Triangle markers indicate that the top value is not significantly higher than the lower one at significance level 0.05 (paired right-tailed t-test).

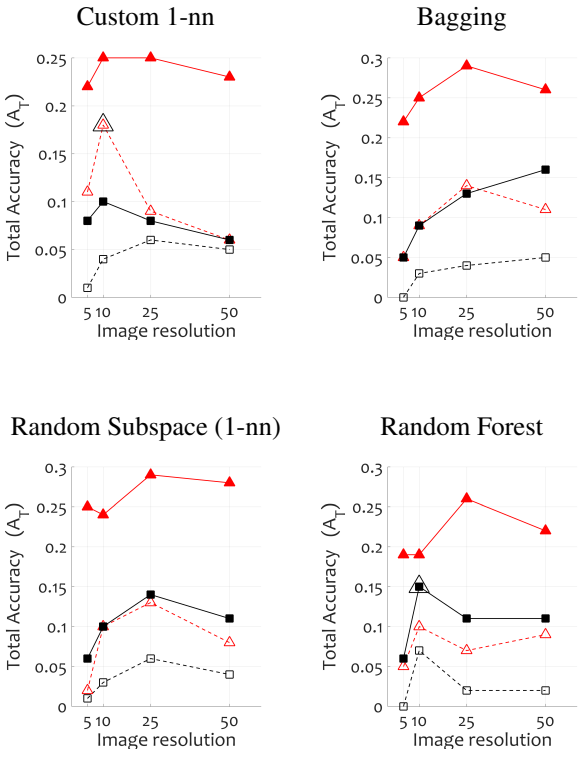

Figure 4: *Total* accuracy A_T for the 4 independent classifier models. Triangle markers indicate no significant difference from the top point at significance level 0.05 (McNemar test).

4.5. Experiments with other feature representations

To strengthen the message that including conditional prior probabilities improves the restricted set classification (RSC), we extracted two additional feature spaces from the image data: the local binary feature descriptors (LBP) [20] and histogram oriented gradients (HOG) [8]. Both feature spaces are orientation-invariant, and very useful descriptors for object's shape and texture. The experimental protocol is the same as with the grey scale intensity data but this time we did not tune specifically the power constant β. We set it at $\beta = 0.1$ for all experiments. Again, the classifier was trained on $\mathcal{D}_2$ and the accuracies A_P and A_T for the four scenarios were calculated on the unseen testing data $\mathcal{D}_3$. The results for A_P and A_T are reported in Tables 2 and 2, respectively. Statistical tests were carried out again to determine whether the accuracies obtained through the proposed method RSC($D + P$) are indeed significantly higher than those of D, $D + P$, and RSC(D). Paired right-tailed t-test was applied for A_P, and McNemar test for A_T. All differences in favour of A_P for both feature spaces were found to be significant. On the other hand, statistical significance was not observed as often for A_T. Still, we note that all values of A_T for RSC($D + P$) were strictly greater than those for the other three methods (apart from LBP and Random subspace where the values are the same).

Overall, LBP feature space was less successful than HOG. HOG showed similar accuracies to the ones obtained with the grey scale intensity features.

This experiment reinforces our observation that including the conditional prior probability improves RSC, and this improvement is not tied up to a serendipitous choice of a specific feature space.

Table 2: Partial accuracy A_P in % for the four scenarios and the two feature spaces. All statistical tests (right-tailed t-test) confirmed that RSC(D+P) is significantly better than the other three scenarios.

CL	FS	D	$D + P$	RSC(D)	RSC(D+P)
LDC	LBP	81.94	88.48	82.50	89.53
	HOG	96.27	96.63	96.80	97.16
1nn	LBP	85.23	87.61	86.17	89.38
	HOG	94.86	95.33	95.08	95.66
BAG	LBP	84.30	89.33	84.98	90.28
	HOG	94.22	95.47	94.52	95.84
RS	LBP	84.38	88.88	85.06	89.81
	HOG	96.22	96.70	96.66	97.06
RF	LBP	84.47	89.33	85.17	90.41
	HOG	93.98	95.38	94.56	95.84

Notes: CL = classifier model, FS = feature space.

4.6. Sensitivity to sample size

The proposed extension would work if there is a substantial data resource from which we can calculate good estimates of the conditional priors $P_P(\omega_i|\theta)$. To evaluate the sensitivity of the method to different data sizes, we carried out the following experiment. The prior probabilities were calculated from data

Table 3: Total accuracy A_T in % for the four scenarios and the two feature spaces. The values found to be not significantly different from RSC(D+P) by the McNemar test are shown in boxes.

CL	FS	D	$D + P$	RSC(D)	RSC(D+P)
LDC	LBP	0.00	0.00	0.00	3.00
	HOG	0.00	8.00	15.00	18.00
1nn	LBP	0.00	0.00	0.00	1.00
	HOG	0.00	5.00	7.00	10.00
BAG	LBP	0.00	0.00	0.00	2.00
	HOG	0.00	6.00	5.00	14.00
RS	LBP	0.00	0.00	0.00	0.00
	HOG	0.00	13.00	13.00	17.00
RF	LBP	0.00	0.00	0.00	2.00
	HOG	0.00	8.00	6.00	13.00

of sizes K taking values: 1000, 5000, 10000, 50000, 100000, 200000, and 300000. We chose the linear discriminant classifier with the HOG feature space as this combination was found to be the most accurate one in Section 4.5. Figure5 shows A_P as a function of K. The quality of the estimates improves and levels off with the size, and so does the classification accuracy A_P of $D + P$ and RSC($D + P$). For D and RSC(D), the accuracy is constant as they do not depend on the estimated prior probabilities. As before, we depict with triangles the accuracies which are not significantly different from A_P of RSC(D) according to the right-tailed paired t-test.

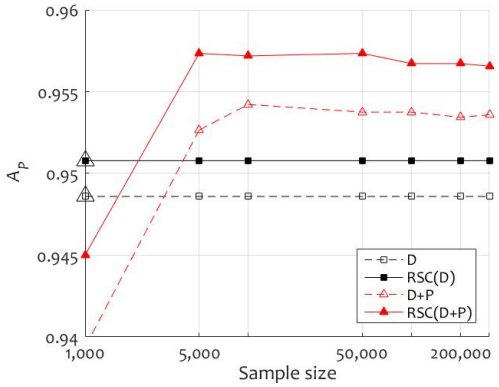

Figure 5: Partial accuracy A_P as a function of the number of instances from which the prior probabilities $P_P(\omega_i|\theta)$ were calculated. The experiments were carried out with HOG features and LDC classifier.

This experiment indicates that in order to take advantage of the RCS extension, we need access to large data or accurate probability estimates obtained from other sources.

5. Conclusions

In this study we extend the Restricted Set Classification (RSC) approach. We propose that including prior probabilities related to the set of instances X being classified together

improves on the overall accuracy of the set classifier. The paper proposes a formal way of including these probabilities and weighting their contribution.

Our hypothesis is tested on a real-life instance of an RSC problem: recognising the pieces on a chessboard from a top-view image. Our experiment demonstrated that including prior probabilities improves significantly the performance of RSC, which in itself is a better solution to the RSC problem compared to an independently applied classifier D. This improvement was found to be robust with respect to the classifier model used as D, and also with respect to the image resolution.

The practical aspect of the case study is that, by using an extended RSC approach, we can achieve a good classification accuracy where this was deemed impossible in the literature: classification of the pieces on a chessboard from a single top-view image, without knowing the game moves leading to the current position, and without marking or modifying the chess board and the pieces in any way.

The extended RSC approach can be applied for any representation of the objects, not only the top-view squares. A side-view sub-image could be added to the top-view representation of each square, or the features extracted from the different images could be concatenated.

It will be interesting to extend the RSC approach further, in at least two directions. First, the posterior probabilities for the independent classifier D can be honed by incorporating evidence from the past success of D in assigning a given label. Second, relationships between the instances in X can be useful for the overall assignment of the labels. For example, suppose that we are recognising the individual students in a class from head-and-shoulder snapshots. Suppose that we know that Peter and John are best friends, and are usually both present or both absent. The posterior probabilities can be altered based on this piece of knowledge.

Acknowledgement

This work was done under project RPG-2015-188 funded by The Leverhulme Trust, UK.

References

[1] Alia, M.A., Tamimi, A.A., AL-Allaf, O.N.A., 2013. Integrated system for monitoring and recognizing students during class session. The International Journal of Multimedia & Its Applications 5, 45–52.

[2] Breiman, L., 1996. Bagging predictors. Machine Learning 26, 123–140.

[3] Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

[4] Chen, A.T.Y., Kevin, I., Wang, K., 2016. Computer vision based chess playing capabilities for the baxter humanoid robot, in: IEEE 2016 The 2nd International Conference on Control, Automation and Robotics (ICCAR 2016), IEEE.

[5] Cohen, I., Goldszmidt, M., 2004. Properties and benefits of calibrated classifiers, in: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 125–136.

[6] Cour, T., Lauranson, R., Vachette, M., 2002. Autonomous chess-playing robot. Undergraduate thesis.

[7] Crouse, D., Jacobs, R.L., Richardson, Z., Klum, S., Jain, A., Baden, A.L., Tecot, S.R., 2017. Lemurfaceid: a face recognition system to facilitate individual identification of lemurs. BMC Zoology 2. doi:DOI:10.1186/s40850-016-0011-9. (open access).

[8] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 886–893 vol. 1. doi:10.1109/CVPR.2005.177.

[9] Danner, C., Kafafy, M., 2015. Visual chess recognition. Stanford University, USA.

[10] Illeperuma, G.D., 2011. Using image processing techniques to automate chess game recording. Proceedings of the Technical Sessions of the Institute of Physics â€“ Sri Lanka 27, 76–83.

[11] Khosla, A., Hamid, R., Lin, C.J., Sundaresan, N., 2013. Large-Scale Video Summarization Using Web-Image Priors. Proc. IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition (2013) , 2698–2705.

[12] Koray, C., Sümer, E., 2016. A computer vision system for chess game tracking, in: In Porc. of the 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia.

[13] Kumar, S., Singh, S.K., 2017. Visual animal biometrics: survey. IET Biometrics 6, 139–156. doi:10.1049/iet-bmt.2016.0017.

[14] Kuncheva, L.I., 2010. Full-class set classification using the Hungarian algorithm. International Journal of Machine Learning and Cybernetics 1, 53–61. doi:DOI10.1007/s13042-010-0002-z.

[15] Kuncheva, L.I., Rodríguez, J.J., Jackson, A.S., 2017. Restricted set classification: Who is there? Pattern Recognition 63.

[16] Matuszek, C., Mayton, B., Aimi, R., Deisenroth, M.P., Bo, L., Chu, R., Kung, M., LeGrand, L., Smith, J.R., Fox, D., 2011. Gambit: An autonomous chess-playing robotic system, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE. pp. 4291–4297.

[17] McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H., 2000. Tracking groups of people. Computer Vision and Image Understanding 80, 42 – 56. doi:http://dx.doi.org/10.1006/cviu.2000.0870.

[18] McNemar, Q., 1947. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12, 153–157.

[19] Neufeld, J.E., Hall, T.S., 2010. Probabilistic location of a populated chessboard using computer vision, in: 2010 53rd IEEE International Midwest Symposium on Circuits and Systems, IEEE. pp. 616–619.

[20] Ojala, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987. URL: http://dx.doi.org/10.1109/TPAMI.2002.1017623, doi:10.1109/TPAMI.2002.1017623.

[21] Piškorec, M., Antulov-Fantulin, N., Ćurić, J., Dragoljević, O., Ivanac, V., Karlović, L., 2011. Computer vision system for the chess game reconstruction, in: MIPRO, 2011 Proceedings of the 34th International Convention, IEEE. pp. 870–876.

[22] Schwenk, A., Yuan, C., 2015. Visual perception and analysis as first steps toward human–robot chess playing, in: International Symposium on Visual Computing, Springer. pp. 283–292.

[23] Technology, D.G., 2017. Electronic boards. URL: {\url{http://www.digitalgametechnology.com/index.php/products/electronic-boards}}. Accessed: 2017-01-21.

[24] Wang, V., Green, R., 2013. Chess move tracking using overhead rgb webcam, in: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), IEEE. pp. 299–304.

[25] Zhang, D., Lu, G., 2002. A comparative study of fourier descriptors for shape representation and retrieval, in: Proc. of 5th Asian Conference on Computer Vision (ACCV), Springer. pp. 646–651.